

Yield-Aware Hierarchical Optimization of Large Analog Integrated Circuits [‡]

Guo Yu and Peng Li

Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX 77843
Email: {yuguo, pli}@neo.tamu.edu

Abstract—Hierarchical optimization using building circuit block pareto performance models is an efficient and well established approach for optimizing the nominal performances of large analog circuits. However, the extension to yield-aware hierarchical methodology, as dictated by the need for safeguarding chip manufacturability in scaled technologies, is completely nontrivial. We address two fundamental difficulties in achieving such a methodology: yield-aware pareto performance characterization at the building block level and yield-aware system-level optimization problem formulation. It is shown that our approach is not only able to effectively capture the block performance trade-offs at different yield levels, but also correctly formulate the whole system yield and efficiently perform system-level optimization in presence of process variations. Our approach extends the efficiency of hierarchical analog optimization, enjoyed for improving nominal circuit performances, to yield-aware optimization. Our methodology is demonstrated by the hierarchical optimization of a phased locked loop (PLL) consisting of multiple circuit blocks.

I. INTRODUCTION

The proliferation of communication and consumer electronic systems leads to high demands for low-power & high-performance ICs. Analog/mixed-signal subsystems of these integrated systems are often the major bottleneck in the whole system design [1]. Automated optimization is a possible solution to this problem, which usually can be classified into two categories: simulation-based and performance-model based techniques. Both types of the approaches can be either conducted in a flat or hierarchical fashion. The latter approach, which decomposes the task of optimizing a complex analog systems into that of several but smaller building blocks, alleviates the optimization cost of large analog systems.

The most widely adopted hierarchical analog optimization approach is based upon modeling the best performance trade-offs (pareto fronts) of competing performances at the building circuit block level [2], [3], [4]. In addition to the optimization of nominal circuit performances, robust analog circuit optimization is becoming increasingly important for modern manufacturing processes [5], [6], [7]. Without considering process fluctuations, automated analog optimization algorithms typically push the system performances to certain process corners that are vulnerable to parametric variabilities [6], [7]. Therefore, in practice we should not only seek for the optimal nominal case performances but also assess the robustness of system performances to safeguard the overall parametric design yields. However, the fulfillment of this need introduces significant complications.

In this paper, we propose a pareto front based hierarchical methodology for yield-aware analog optimization. To authors' best knowledge, there does not exist any prior method that effectively achieves the same goal. Our main contributions come from solving

two basic difficulties in facilitating such a hierarchical optimization framework. First, we take a new look at the building block level performance pareto front generation and introduce a new form of pareto fronts while still maintaining the benefit of reduced search space for system optimization. Second, we address the issue of estimating the whole system performance distribution, which is a key component of the system-level optimization objective functions. The system yield estimation under the hierarchical framework is hampered by the following difficulty: while performance yields can be characterized at the building block level, the whole system yield cannot be correctly estimated using the yields of individual blocks. This complication is introduced by correlated variation sources that impact multiple building blocks and hence introduce unavoidable correlations between them. We remedy this problem by introducing a new system-level yield-aware optimization formulation. The formulation correctly estimates the whole system yield, thereby bridging the block level and the system-level under the context of robust optimization.

II. BACKGROUND

A. Hierarchical Optimization in Nominal Case

In hierarchical optimization, a large analog system is decomposed into several building blocks. In order to get the best overall system performances, it is natural to find the design points which result in best performances for the building blocks. For most of circuits blocks, different performance objectives compete against each other and it is infeasible to find a design point to reach the best value for all performance objectives at the same time. The design task then becomes a multi-objective optimization problem which is to find the best performance trade-offs (pareto fronts). In multi-objective optimization, performance p_a dominates performance p_b (suppose smaller value is better) when [8], [7]

$$\mathbf{p}_a \prec \mathbf{p}_b : \forall (p_{a_i} \leq p_{b_i}) \wedge \exists (p_{a_i} < p_{b_i}), \quad i = 1, \dots, n \quad (1)$$

where p_{a_i} and p_{b_i} are the i -th performances of interest, and there are totally n performances. A set of performances is considered as pareto-optimal if it is not dominated by any other set of performances.

The obtained pareto fronts represent the best performance trade-offs the circuit blocks can achieve. Then, the optimal system performances can be found by searching within these building block-level pareto fronts. In particular, circuit level design parameters (transistor sizes, passive components, biasing, etc) are explored within the design constrains to find the best possible performance trade-offs. The system level optimization is carried out by searching in the space constrained by block-level pareto fronts. There exist two key benefits for this hierarchical optimization. First, since the number of performances in the block level is much smaller than that of the original design space, the search space can be reduced significantly, leading to improved optimization efficiency. An equally important

[‡] This work was funded in part by the FCRP Focus Center for Circuit & System Solutions (C2S2), under contract 2003-CT-888.

benefit here is that system-level behavioral models can be used to quickly estimate system-level performances, thereby further reducing the overall optimization cost.

B. Kriging Performance Model

Pareto fronts can be generated using simulation-based method or performance models [5], [7], [9]. In order to alleviate the optimization cost, we use the performance model based approach in the proposed hierarchical yield-aware optimization flow. While performance models may be extracted in various ways, we have found a geostatistics motivated technique, Kriging model [10], is particularly interesting. As a nonlinear regression tool, Kriging model not only provides circuit performance predictions, but also offers a confidence level for each prediction. This feature favorably allows for an iterative update approach for the extraction of high-dimensional performance models, which is demanded in analog circuit optimization [9].

In the proposed optimization flow, Kriging modeling is used for two different purposes. First, parameterized Kriging models in both the design parameters and process variables are extracted to predict statistical building block performances. In cases where there are a large number of process variables, parameter dimension reduction methods can be additionally applied to reduce the number of process variables to simplify the Kriging model extraction [11]. These block-level Kriging models are used in an iterative optimization process to generate yield-aware block-level (multi-yield) pareto fronts. In addition, Kriging modeling can also be used to efficiently map block-level performances (e.g. system-level model parameters) to system-level performances. However, it should be noted that proposed hierarchical statistical system optimization approach does not depend on this choice of performance modeling technique. Therefore, Kriging modeling is treated as a “black-box” tool and in principle it can be replaced by any other suitable modeling technique.

III. ISSUES IN HIERARCHICAL YIELD-AWARE OPTIMIZATION

Hierarchical yield-aware optimization presents several challenges. In the nominal case, circuit blocks can be individually characterized in terms of block-level pareto performance models and then the system-level optimization can be done in terms of the block-level best performance tradeoffs. The use of pareto modeling provides a clean interface between different blocks, enabling efficient hierarchical optimization. However, in the case of yield-aware optimization, it is critical to capture the impacts of device-level variations on the system-level performances. This need makes individual extraction of block-level pareto models, as needed in the hierarchical optimization, much more involved.

A. Block-level Issues

Different methods for yield-aware pareto front generation have been proposed [5], [7], [9]. To obtain yield-aware pareto models, block-level design points that provide robust best block-level performance tradeoffs are collected. This is achieved by assuming a single yield target for possibly multiple block-level performances. However, there exists a disconnection between the block-level models and the system-level performances, where statistical variations are considered at the system level. In other words, the single yield level at the block level is not sufficient to provide enough statistics based on which the whole system yield can be estimated.

B. System-level Issues

In [7], the authors suggested to use the building block pareto fronts with all the performances at one specified yield level in the hierarchical optimization. And the obtained system-level optimization

points were supposed to have the same yield level as building blocks. This approach, however, may not work properly due to several reasons.

Firstly, the transformations from building block-level performances to system-level performances may be complex. This dependency may allow the low yield level of one building block be compensated by other blocks in the same system. Similarly, the performances within one building block may also be compensated by the block-to-system transformation. As a result, the relationship between building block yields and the system yields can be non-monotonic and complex. Only using the specified yield level pareto fronts in the hierarchical optimization will lose many possible promising block-level performance combinations which may lead to better system-level performances in the end.

The severer problem comes from the statistical correlations between various circuit blocks. In reality, the device variations in different blocks may share common/global physical origins. As a result, not only the device variations are correlated, so are the block-level performances across the blocks. In the prior yield aware pareto front modeling works [5], [7], [9], such correlations are not captured since each block is optimized independently. This issue is especially severe if a single yield target is assumed for all the block-level performances when the pareto models are extracted. Such a *simple* yield-aware pareto model can not provide full statistical information to determine the whole system yield. Consider a simple example, where the entire system consists of two blocks with two block-level performances P_1 and P_2 . And the system performance is simply assumed to be: $P_s = P_1 + P_2$. If P_1 and P_2 are of Gaussian distribution then the system P_s will also be Gaussian. In [7], it is suggested that to achieve a system-level yield target, say 84.1%, the block-level pareto models at the same yield level should be considered. In this simple case, the system performance that achieves the yield target is at: $\mu_1 + \mu_2 + \sqrt{\sigma_1^2 + \sigma_2^2 + 2\sigma_1\sigma_2 \cdot \text{cov}(P_1, P_2)}$. Obviously, the value for this performance level depends on the correlation between P_1 and P_2 . Without such knowledge, the correct system-level performance cannot be decided. The situation becomes even more complex if the block performance distributions are non-Gaussian. In this case, knowing only the correlation factor is also not sufficient. We address these challenges by using the techniques described in the following sections.

IV. MULTI-YIELD PARETO FRONTS

Unlike the prior work where a single fixed yield level is used when extracting the yield-aware pareto front for multiple performances [5], [7], [9], we introduce the notation of *multi-yield pareto fronts*, where best performance trade-offs are extracted in terms of combinations of yield level parameters individually specified for each performance. For example, a block with two performances P_1 and P_2 , will be characterized in terms of two sperate yield level parameters Y_1 and Y_2 , one for each performance.

A. Model Generation

The multi-yield pareto fronts are generated by varying the yield level for each performance individually. For the i -th building block our proposed multi-yield pareto front is in the form of

$$\begin{aligned} MY(\vec{Y}_{Bi}, \vec{P}_{Bi}) &= 0 \\ \vec{Y}_{Bi,Min} &\leq \vec{Y}_{Bi} \leq \vec{Y}_{Bi,Max} \end{aligned} \quad (2)$$

where $\vec{P}_{Bi}$ are the best block performances that can be achieved at the yield level $\vec{Y}_{Bi}$. For practical purpose, $\vec{Y}_{Bi}$ is constrained within $[\vec{Y}_{Bi,Min}, \vec{Y}_{Bi,Max}]$. The yield level $\vec{Y}_{Bi}$ can vary for different

performances in a building block. An example of two-performance multi-yield pareto front generation is shown in Fig. 1. In this case, a fixed-yield pareto model is extracted at each combination of the two performance yield targets.

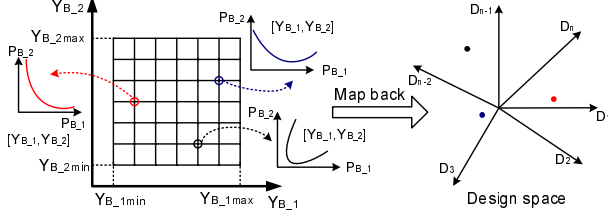


Fig. 1. Multi-yield pareto front generation.

The use of multi-yield pareto fronts allows us to identify a more complete set of “near optimal” block-level design points for hierarchical optimization. Facing the lack of the system-level interaction during the individual pareto model extraction stage, this choice relaxes the artificial constraints set in the fixed-yield pareto models and allows the system-level optimization be conducted in a larger number of promising block level performance tradeoffs. However, the search space for the system-level optimization are still constrained by the block-level best performances trade-offs. Therefore, the hierarchical nature of the overall optimization is preserved.

B. Link to the System-level Optimization

As described before, capturing statistical correlations between various block-level performances is essential for the system-level analysis and optimization. This can be achieved using the multi-yield pareto models as follows. In order to evaluate the system performance distribution correctly, we need the performance distributions of building blocks for all possible promising design points. As such, the yield levels in multi-yield pareto fronts are not used directly but together with block performances to identify the design parameters. A specified yield level and performance set $(\vec{Y}_{Bi}, \vec{P}_{Bi})$ can identify a unique design point in the multi-yield pareto front of the i -th building block. When appropriate, interpolation along the pareto front can be conducted. The mapping to the design space of the i -th building block can be denoted as

$$\vec{D}_{Bi} = D_{PY_i}(\vec{Y}_{Bi}, \vec{P}_{Bi}). \quad (3)$$

This mapping can be achieved by using Kriging performance models to generate dense points forming pareto fronts instead of analytical formulas, so the mapping back to the design parameters is naturally obtained.

V. YIELD-AWARE HIERARCHICAL OPTIMIZATION

The N block-level multi-yield pareto fronts are parameterized in the block-level yield targets and the corresponding achievable performances $[\vec{Y}_B, \vec{P}_B]$, where $\vec{Y}_B = [\vec{Y}_{B1}, \dots, \vec{Y}_{BN}]$ and $\vec{P}_B = [\vec{P}_{B1}, \dots, \vec{P}_{BN}]$. In the following, the system-level optimization is performed using these multi-yield pareto fronts.

A. System-level Optimization Cost Function

The goal of the yield-aware circuit optimization is to find the optimal system performances at targeted system yields. There can be more than one performances for the whole system. Therefore, we can also specify different system yield levels for different performances. Due to process variations, the system performances are all statistical variables. For the k -th statistical system performance $P_{s,k}$ (smaller

the better), suppose we need a yield of $Y_{s,k}$, then the yield-aware performance $P_{s,k}^{Y_{s,k}}$ satisfies the following probability condition

$$P\{P_{s,k} \leq P_{s,k}^{Y_{s,k}}\} = Y_{s,k}. \quad (4)$$

Eqn. 4 implies that for $P_{s,k}$, the best achievable performance value is $P_{s,k}^{Y_{s,k}}$ when yield level $Y_{s,k}$ is required. $P_{s,k}^{Y_{s,k}}$ is considered as the yield-aware k -th system performance and to be used in the system-level optimization.

For multi-objective systems, the system-level cost function F for M system performances with yield $Y_s = [Y_{s,1}, \dots, Y_{s,M}]$ can be formulated as

$$F(\vec{Y}_B, \vec{P}_B) = \sum_{k=1}^M \frac{W_{s,k} \cdot P_{s,k}^{Y_{s,k}}(\vec{Y}_B, \vec{P}_B)}{Spec_k} \quad (5)$$

where $W_{s,k}$ is the weighting coefficient for the k -th system performance and $Spec_k$ is its specified performance achievable at yield level $Y_{s,k}$. The input variables for the cost function are the yield and performance set $[\vec{Y}_B, \vec{P}_B]$ of all the building blocks. With Eqn. 5, the objective of yield-aware optimization is to minimize the cost function $F(\vec{Y}_B, \vec{P}_B)$ at specified system yield levels $Y_{s,k}$. By changing the weighting coefficients W_s for different system performances, the system-level optimization can be set to tradeoff between different system performances.

B. Performance Evaluation

The system cost function has been formulated. Now the question is how to optimize it. As we are interested in the system performances in the statistical sense, accurate evaluation of statistical system performances is required. Since the optimization variables are $[\vec{Y}_B, \vec{P}_B]$, we need to know the mapping function $f_{s,k}$ to get $P_{s,k}^{Y_{s,k}}$ from the multi-yield pareto fronts, formulated as $P_{s,k}^{Y_{s,k}} = f_{s,k}(\vec{Y}_B, \vec{P}_B)$.

To achieve this, we first transfer the optimization variables in each building block back to the design space using Eqn. 3, then we perform Monte-Carlo simulation at each design point to evaluate the block performance distributions. This step can be accelerated by extracting an empirical Kriging based regression model. If there are correlations between device-level parameters across different building blocks, they can be naturally captured in the block performance distributions as it is now possible to generate Monte-Carlo samples at the block level with such correlations considered. The correlated block performance distributions are then mapped into the system performance distributions using system-level behavioral simulation. Again, if needed, this step can be spedup by extracting a Kriging regression model. The system performances at yield Y_s are obtained by finding the values meeting the yield requirement in the system performance distributions. The mapping flow from multi-yield pareto fronts to system performances is shown in Fig. 2.

C. Optimization Formulation

The complete flowchart of the hierarchical optimization using multi-yield pareto fronts is illustrated in Fig. 3. We start from the yields and performances of multi-yield pareto fronts, get back to the design points, then obtain the statistical system performances to evaluate the system cost function. The optimization goal is to reduce the cost function in Eqn. 5. Since the multi-yield pareto fronts are self constrained, the optimizer also need to take Eqn. 3 as the optimization constrain.

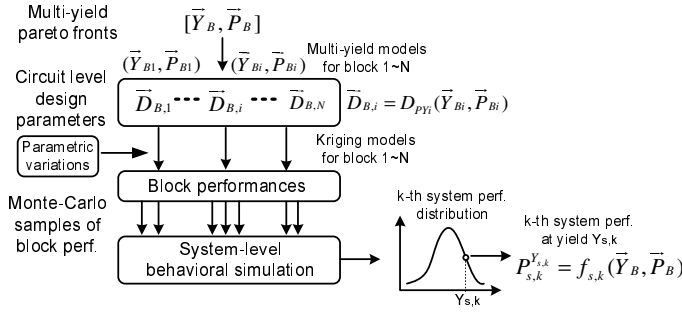


Fig. 2. Mapping from multi-yield pareto fronts to yield-aware system performances.

For a system with M performances and N building blocks, the complete system-level optimization can be formulated as

$$\begin{aligned} \min & F(\bar{Y}_B, \bar{P}_B) \\ \text{s.t.} & \begin{cases} P_{s,k}^{Y_{s,k}} = f_{s,k}(\bar{Y}_B, \bar{P}_B), & k = 1, 2, \dots, M \\ MY(\bar{Y}_{Bi}, \bar{P}_{Bi}) = 0, & i = 1, 2, \dots, N \\ \bar{Y}_{Bi-Min} \leq \bar{Y}_{Bi} \leq \bar{Y}_{Bi-Max} \end{cases} \end{aligned} \quad (6)$$

where the optimization variables are multi-yield pareto front yields and performances $[\bar{Y}_B, \bar{P}_B]$ for all the N building blocks. The dimensions of $\bar{Y}_{Bi}$ and $\bar{P}_{Bi}$ in each building block are the same, which depend on the number of block-level performances considered. Note that each multi-yield pareto front acts as an constraint in the optimization and reduces the degrees of freedom for of the system-level optimization by one. Suppose the i -th block of has L_i block-level performances, the dimension of the optimization search space is $\sum_i^N (2L_i - 1)$. Hence, the proposed approach extends the efficiency of hierarchical analog optimization from deterministic optimization to statistical optimization.

The system-level optimization problem can be solved by any suitable optimization method, particularly a derivative free method. In this paper, global optimization algorithm based on multilevel coordinate search (MCS) [12] is adopted.

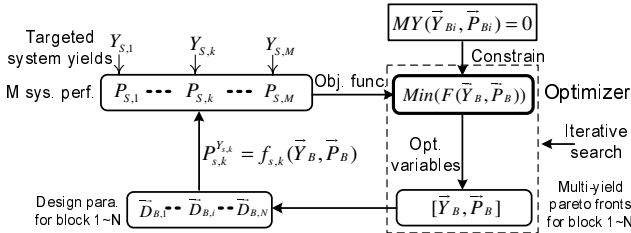


Fig. 3. Hierarchical optimization using multi-yield pareto fronts.

VI. EXPERIMENTAL RESULTS

We demonstrate the detailed application of the yield-aware hierarchical optimization including behavioral modeling, multi-yield pareto front generation and system-level optimization formulation for a charge-pump PLL. Due to the mixed-signal nature, the design and evaluation of a PLL system is quite complex and costly. As a result, the brute-force optimization by searching in the design space with transistor-level simulation is often infeasible.

A. Behavioral Model Generation

The PLL system investigated contains a voltage control ring oscillator, a phase detector, a charge pump, a loop filter and a

frequency divider. Among these components, the phase detector and the frequency divider are digital components that are rather robust to process variations. Hence, we focus on the optimization of VCO, charge pump and filter in this example.

The performances considered in the VCO behavioral model are *jitter*, *power*, VCO gain K_{VCO} and oscillation frequency F_{offset} at certain control voltage V_{offset} . The VCO voltage-frequency curve is linearized around V_{offset} with the slope of K_{VCO} . V_{offset} is predefined as the center of VCO linear gain region. When the control voltage is too low ($V_{in} < V_{low}$) or too high ($V_{in} > V_{high}$), the VCO gain starts to saturate. These two nonlinear gain regions are also modeled, as illustrated in Fig. 4. The VCO voltage-to-frequency model is then constructed as

$$Freq = \begin{cases} K_{high} \cdot (V_{in} - V_{high}) + F_{high} & V_{in} > V_{high} \\ K_{low} \cdot V_{in} & V_{in} < V_{low} \\ K_{VCO} \cdot (V_{in} - V_{offset}) + F_{offset} & \text{otherwise} \end{cases} \quad (7)$$

Here the predefined V_{low} and V_{high} voltages set the two gain saturation regions in the VCO curve. K_{low} , K_{VCO} and K_{high} are the VCO's gains in the three piecewise linear regions, and F_{high} and F_{offset} are certain constants. The VCO behavior as in Eqn. 7 can capture the complete voltage-frequency transfer curve, which is needed model to the PLL acquisition behavior. In literature, simpler fixed-gain based VCO models are used in PLL optimization frequency region [3], [13]. In addition to modeling the VCO nonlinear voltage-to-frequency characteristics as in Eqn. 7, when appropriate, other detailed circuit level characteristics can be also introduced under our hierarchical optimization framework. For the charge pump, we include *jitter*, charge up current I_{up} and charge down current I_{down} in the behavioral model. The parameters of loop filters are capacitances C_1 and C_2 , resistance R (Fig. 4), which are all handled directly in system-level simulation. The power of charge pump and loop filter can be calculated with charge pump currents [3]. The behaviors of digital building blocks in the system, including frequency divider and the phase detector, are also characterized using delay and slew and included in the behavioral models.

After the behaviors of analog building blocks are extracted, they are mapped into Verilog-A models for the system-level simulation [3], [14]. At the system level, the performances of the PLL considered are lockin time T , power P (exclude digital blocks) as well as jitter J [15]. The modeling and simulation of the PLL using behavior models are illustrated in Fig. 4. With the Verilog-A behavioral models, the mapping from block-level performances to the system-level performances can be achieved efficiently.

B. System Level Optimization

Multi-yield pareto fronts for individual building blocks are extracted to perform yield-aware hierarchical optimization. For the VCO, the VCO behavioral model parameters K_{low} , K_{high} , V_{low} , V_{high} , F_{offset} and F_{high} are not considered as performance metrics for the pareto model generation. The pareto fronts in the form of $P_{VCO}(\text{power}, \text{jitter}, \text{gain})$, where *gain* corresponds to K_{VCO} , are extracted by finding the best tradeoffs between the three performance metrics. Other behavioral model parameters are retrieved from the corresponding design information on the pareto fronts. For the charge pump, ideally the charge up and down currents (I_{up} and I_{down}) shall be identical. However, the mismatch between them may be introduced by process variation, impacting the PLL jitter and lockin time. The pareto fronts of the charge pump are extracted considering jitter, average charge current $I_{cp} = 0.5 \cdot (I_{up} + I_{down})$ and mismatch current $I_{mis} = \text{abs}(I_{up} - I_{down})$: $P_{CP}(\text{jitter}, I_{cp}, I_{mis})$. The

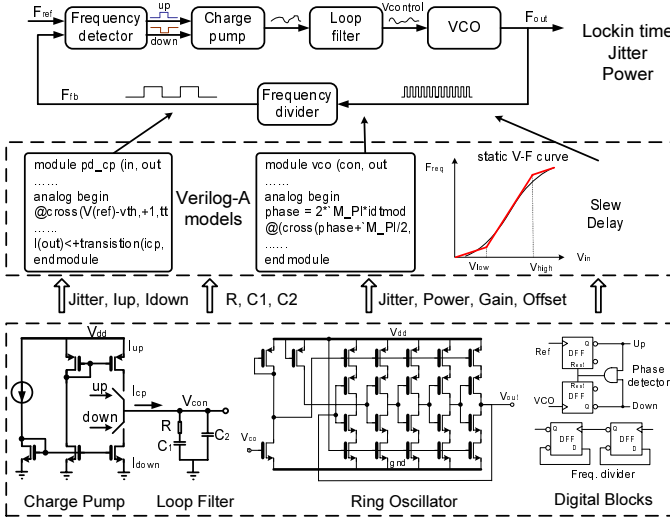


Fig. 4. PLL hierarchical modeling and simulation.

power consumption is not included in the pareto fronts since it is proportional to I_{cp} .

For the PLL optimization, the cost function (Eqn. 5) is

$$F(\vec{Y}_B, \vec{P}_B, R, \vec{C}) = W_P \cdot \frac{P_{PLL}(\vec{Y}_B, \vec{P}_B, R, \vec{C})}{P_{Spec}} + W_J \cdot \frac{J_{PLL}(\vec{Y}_B, \vec{P}_B, R, \vec{C})}{J_{Spec}} + W_T \cdot \frac{T_{PLL}(\vec{Y}_B, \vec{P}_B, R, \vec{C})}{T_{Spec}} \quad (8)$$

with the constraints (see also Eqn. 6)

$$\begin{cases} MY_{CP}(Y_{jit}, P_{jit}, Y_{Icp}, P_{Icp}, Y_{Imis}, P_{Imis}) = 0 \\ MY_{VCO}(Y_{jit}, P_{jit}, Y_{pow}, P_{pow}, Y_{gain}, P_{gain}) = 0 \\ R_{min} \leq R \leq R_{max}; \vec{C}_{min} \leq \vec{C} \leq \vec{C}_{max}; \vec{Y}_{min} \leq \vec{Y} \leq \vec{Y}_{max} \end{cases} \quad (9)$$

where W_P , W_P , W_J , P_{spec} , W_{spec} and J_{spec} are the weighting factors and the specified performance targets achievable at certain yield levels for the three system performances. In the above formulation, the optimization variables include not only the yield-levels and performances for the VCO and charge pump multi-yield pareto fronts, but also the loop filter parameters (R , $\vec{C}$). In our implementation, the latter set of variables are considered as deterministic and bounded.

C. Optimization Results

The charge-pump PLL is implemented in 90nm CMOS technology. The process variations considered include the threshold voltage V_{th} for each transistor with a variation of $3\sigma=10\%$. For the system-level simulation, a single transient simulation of $10\mu s$ for jitter and lockin time analysis requires about 40 seconds even with Verilog-A models. To alleviate the optimization cost, we build another Kriging model to map all the building block-level performances (VCO_{power} , VCO_{jitter} , etc) and filter parameters to the PLL system performances, then use these high-level Kriging models to guide the optimizer to find optimal solutions. After the optimal design points are obtained, Verilog-A based simulations are used to find the performances corresponding to specified yields in these points.

To verify the accuracy of the Kriging based block-level performance modeling, four random design points are selected and at these design points the VCO Kriging model is compared with transistor-level transient simulation. As shown in Table I, the Kriging model predicts the VCO gain, power, offset and jitter rather accurately

as compared with transistor level transient simulation. Our Kriging based multi-yield pareto fronts for the charge pump and VCO are shown in Fig. 5. To verify the accuracy of the modeling based PLL system performance evaluation, in Table II, the nominal performances at one design point are examined. First, the complete PLL is simulated using flat transistor-level transient simulation and the simulated performances are listed in the first row. As a comparison, the system performances are more efficiently evaluated using Verilog-A system-level simulation and the results are shown in the second row. In the latter case, the model parameters of the system-level Verilog-A models are also predicted using the block-level Kriging performance models. It can be seen that the model based results match those obtained from direct transistor-level simulation very well.

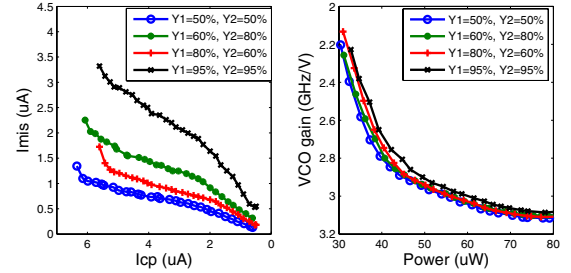


Fig. 5. Multi-yield pareto fronts for charge pump (left) and VCO (right).

Via the proposed system-level optimization, the lockin time and power trade-offs for the PLL are analyzed. As the weighting coefficients for power and lockin time are varied in Eqn. 8, various optimal PLL performance combinations are obtained, as shown in Fig. 6. For illustration purpose, the yields of the two system performances are set to be identical. The performance trade-offs without consideration of yield are obtained in a way similar to the approach in [3], also shown in Fig. 6. We can see from the figure that the yield could be very low if the optimization is performed deterministically, especially for the region around the point denoted as ‘‘Opt. point’’. The nominal system performances of ‘‘Opt. point’’ and the initial design evaluated using direct transistor-level transient simulation are also plotted in Fig. 6, denoted as ‘‘Init. design’’ and ‘‘Opt. design’’, respectively. Similarly, the trade-offs between jitter and lockin time at different yield levels are plotted in Fig. 7.

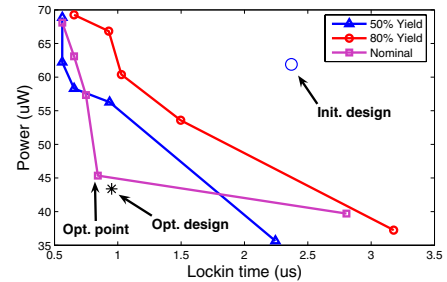


Fig. 6. Trade-offs between lockin time and power at different yield levels.

We verify the optimized tradeoff between power and jitter at 80% yield level in Fig. 8. Since the transistor-level simulation of the PLL is very expensive, we simulate the PLL system performances at a set of 200 randomly selected design points using Verilog-A models. It can be seen that optimized performance trade-off curve in Fig. 8 is superior to all the verification points, confirming the

TABLE I
VERIFICATION OF KRIGING MODEL ACCURACY.

VCO	Gain(model)	Gain(sim.)	Pow(model)	Pow(sim.)	Offset(model)	Offset(sim.)	Jit(model)	Jit(sim.)
Point 1	2.38 GHz/V	2.36 GHz/V	48.1 μ W	48.2 μ W	1.51 GHz	1.54 GHz	1.20 ps	1.16 ps
Point 2	2.91 GHz/V	2.99 GHz/V	55.6 μ W	55.5 μ W	2.58 GHz	2.60 GHz	1.09 ps	1.01 ps
Point 3	3.05 GHz/V	3.09 GHz/V	72.4 μ W	72.5 μ W	2.50 GHz	2.49 GHz	0.88 ps	0.85 ps
Point 4	2.69 GHz/V	2.75 GHz/V	98.9 μ W	99.2 μ W	1.97 GHz	2.00 GHz	0.75 ps	0.76 ps

TABLE II
VERIFICATION OF MODELING BASED PLL SYSTEM PERFORMANCE EVALUATION.

	PLL		VCO				Charge pump			Loop filter		
	Power	Lockin	Gain	Offset	Power	Jitter	I_{up}	I_{down}	Jitter	C1	C2	R
transistor-level	43.4 μ W	0.95 μ s	2.36G/V	1.36GHz	36.0 μ W	0.92ps	3.55 μ A	3.53 μ A	63.0fs	1.5pF	0.36pF	184K
model	45.4 μ W	0.84 μ s	2.39G/V	1.33GHz	36.7 μ W	1.18ps	3.64 μ A	3.62 μ A	68.6fs	1.5pF	0.36pF	184K

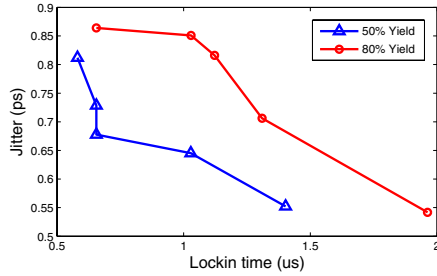


Fig. 7. Trade-offs between lockin time and jitter at different yield levels.

TABLE III
RUNTIME OF HIERARCHICAL YIELD-AWARE OPTIMIZATION.

	VCO	Charge pump
Kriging model gen.	12 min	5 min
Multi-yield pareto gen.	103 min	117 min
Cost function optimization	10 min/point	

effectiveness of the proposed hierarchical yield-aware optimization. The runtime of the proposed PLL optimization is listed in Table III.

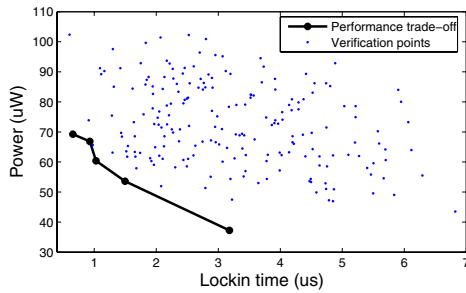


Fig. 8. Verification of performance trade-offs for lockin time and power.

The optimization takes about 10 minutes for a given cost function after all the models are extracted. One complete optimization run including model generation time takes only a few hours, which is quite acceptable for large analog circuit synthesis.

VII. CONCLUSION

In this paper we propose a methodology to perform yield-aware hierarchical optimization by addressing the problems of generating

yield-aware pareto fronts of building blocks and formulating system performances at specified yield levels. The system-level performance distributions are naturally captured by searching in the multi-yield pareto fronts while still maintaining the efficiency of hierarchical optimization. The proposed methodology is validated through the optimization of a large charge-pump PLL design optimization with good efficiency achieved.

REFERENCES

- [1] G. Gielen and R. Rutenbar. Computer-aided design of analog and mixed-signal integrated circuits. *Proc. of the IEEE*, 88(12):1825–1852, Dec. 2000.
- [2] I. Das and J. Dennis. Normal-boundary intersection: A new method generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, pages 631–657, 1998.
- [3] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann. A CPPLL hierarchical optimization methodology considering jitter, power and locking time. In *Proc. of IEEE/ACM DAC*, pages 19–24, July 2006.
- [4] T. Eeckelaert, R. Schoofs, G. Gielen, M. Steyaert, and W. Sansen. Hierarchical bottom-up analog optimization methodology validated by a delta-sigma A/D converter design for the 802.11 a/b/g standard. In *Proc. of IEEE/ACM DAC*, pages 25–30, June 2006.
- [5] B. Smedt and G. Gielen. HOLMES: Capturing the yield - optimized design space boundaries of analog and RF integrated circuits. In *Proc. of IEEE/ACM DATE*, pages 19–24, July 2003.
- [6] X. Li, P. Gopalakrishna, Y. Xu, and L. Pileggi. Robust analog/rf circuit design with projection-based posynomial modeling. In *Proc. of IEEE/ACM ICCAD*, pages 855–862, Nov. 2004.
- [7] S. Tiwary, P. Tiwary, and R. Rutenbar. Generation of yield-aware pareto surfaces for hierarchical circuit design space exploration. In *Proc. of IEEE/ACM DAC*, pages 31–36, July 2006.
- [8] G. Stehr, H. Braeb, and K. Antreich. Performance trade-off analysis of analog circuits by normal-boundary intersection. In *Proc. of IEEE/ACM DAC*, pages 958–963, June 2003.
- [9] G. Yu and P. Li. Yield-aware analog integrated circuit optimization using geostatistics motivated parametric failures. In *Proc. of IEEE/ACM ICCAD*, pages 464–469, Nov. 2007.
- [10] J. Sacks, W. Welch, T. Mitchell, and H. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, Nov. 1989.
- [11] W. Dong, Z. Feng, and P. Li. Efficient VCO phase macromodel generation considering statistical parametric variations. In *Proc. of IEEE/ACM ICCAD*, pages 874–878, Nov. 2007.
- [12] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, June 1999.
- [13] S.K. Tiwary, S. Velu, and R.A. Rutenbar and T. Mukherjee. Pareto-optimal modeling for efficient PLL optimization. *Nanotech*, 2:195–198, 2004.
- [14] Open Verilog International. *Verilog-A Language Reference Manual*. www.verilog.org.
- [15] K. Kundert. *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers*. www.designers-guide.org.